Claims

1. A signal processor for Fast Fourier Transformation, FFT, of M_R, $M_R > 1$, input data streams (x1(n), ..., x4(n)) supplied in parallel,
comprising
- a multiplexing device (MUX) having
  M_R input terminals each receiving one of the M_R input data streams (x1(n), ..., x4(n)) and
  an output terminal (x'(n)) at which the M_R input data streams are output in a multiplexed manner,
- a Fast Fourier Transformation device (FFT)
  configured to perform Fast Fourier Transformation of a data stream supplied at an input terminal (x'(n)) thereof and to output the FFT transformed data stream at an output terminal (X(k)) thereof,
  the input terminal of the Fast Fourier Transformation device (FFT) being connected to the output terminal (X(n)) of the multiplexing device (MUX), and
- a demultiplexing device (DEMUX) having
  an input terminal connected to the output terminal (X(k)) of the Fast Fourier Transformation device (FFT) and
  M_R output terminals (X1(k), ..., X4(k)) at which a respective one of M_R transformed output data streams is output in a demultiplexed manner,

characterized in that

- each of the M_R input data streams contains a number of $N=2^k$ samples,
- the Fast Fourier Transformation device (FFT)
  has a pipeline architecture composed of k stages with a respective feedback path including a single delay element per each stage of the pipeline architecture and
  is controlled by a first (clk') and second internal control signals (s', t', w'),
- wherein

the delay element in a feedback path of an i^{th} stage, $1<=i<=k$, of the pipeline architecture imposes a delay of $M_R*N/2^i$ samples,

the first internal control signal (clk') is clocked M_R times faster compared to a clock rate (clk) at which the samples of the M_R streams are supplied, and

the second internal control signals (s', t', w') are clocked M_R times slower compared to the first internal control signal (clk').

2. A signal processor according to claim 1, wherein

the multiplexing device (MUX) is configured such that the M_R input data streams are multiplexed per data sample of the input data streams and

the demultiplexing device (DEMUX) is configured such that the transformed input data stream is demultiplexed per data sample of the transformed data stream.

3. A signal processor according to claim 2, wherein

a control signal supplied to the multiplexer and demultiplexer is clocked at a rate M_R times the clock rate of the supplied streams.

4. A signal processor according to claim 1, wherein

the Fast Fourier Transformation device (FFT) has a Radix-2 Single-path Delay Feedback, R^2SDF, architecture.

5. A signal processor according to claim 4, wherein

the pipeline architecture of the Fast Fourier Transformation device is composed of Butterfly stages of types I and II (BF2I, BF2II).

6. A signal processor according to claim 5, wherein

the first stage of the pipeline architecture receiving the multiplexed data streams is a Butterfly stage of type I for even and odd total numbers of k.

7. A network element of a communication network comprising a signal processor according to any of the preceding claims 1 to 6.

8. A terminal configured to communicate via a communication network, the terminal comprising a signal processor according to any of the preceding claims 1 to 6.

9. A system comprising at least one of a terminal according to claim 8 and a network element according to claim 7.

10. A signal processing method for performing Fast Fourier Transformation, FFT, of M_R, $M_R > 1$, input data streams $(x1(n), \ldots, x_{M_R}(n))$ supplied in parallel,
comprising the steps of
- multiplexing the M_R input data streams $(x1(n), \ldots, x_{M_R}(n))$ to a multiplexed data stream,
- performing Fast Fourier Transformation of the multiplexed data stream and outputting the transformed data stream,
- demultiplexing the transformed data stream to M_R transformed output data streams,
characterized by
- each of the M_R input data streams contains a number of $N=2^k$ samples,
- performing FFT transformation using a pipeline of k stages with a respective feedback path imposing a delay on the samples per each stage of the pipeline and
- controlling the performing of the FFT transformation by a first (clk') and second internal control signals (s', t', w'),
- and by

imposing a delay of $M_R*N/2^i$ samples on the samples in the feedback path of an i^{th} stage, $1<=i<=k$, of the pipeline,

clocking the first internal control signal (clk') M_R times faster compared to a clock rate (clk) at which the samples of the M_R streams are supplied, and

clocking the second internal control signals (s', t', w') M_R times slower compared to the first internal control signal (clk').

11. A method according to claim 10, wherein

multiplexing is accomplished such that the M_R input data streams are multiplexed per data sample of the input data streams and

demultiplexing is accomplished such that the transformed data stream is demultiplexed per data sample of the transformed data stream.

12. A method according to claim 11, wherein

clocking to the multiplexer and demultiplexer is performed at a rate M_R times the clock rate of the supplied streams.

13. A method according to claim 10, wherein

the Fast Fourier Transformation processing is based on a Radix-2 Single-path Delay Feedback algorithm.

14. A method according to claim 13, wherein

the pipeline of processing stages for the Fast Fourier Transformation is composed of Butterfly stages of types I and II (BF2I, BF2II).

15. A method according to claim 14, wherein

the first stage of the pipeline receiving the multiplexed data stream is a Butterfly stage of type I for even and odd total numbers of k.

16. A computer chip comprising at least a signal processor according to any of the preceding claims 1 to 6.

17. A computer program product for a computer, comprising software code portions for performing the steps of any one of claims 10 to 15 when the program is run on the computer.

18. The computer program product according to claim 17, wherein the computer program product comprises a computer-readable medium on which the software code portions are stored.